



XDI-SWORD-IMG-004

Lab4: 图像处理滤波器实验

1: 均值滤波

Joseph Xu

2018-4-27

修改记录


版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-4-27

审核记录

姓名	职务	签字	日期

目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	6
1.1 概述.....	6
1.2 实验目标.....	7
1.3 实验条件.....	7
1.4 实验原理.....	7
2. 均值滤波实验流程.....	12
2.1 操作步骤.....	12
3. 均值滤波实验结果.....	31

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	2 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

图目录

图 1-1	实验连接示意图.....	8
图 1-2	图像对比度调节连接示意图.....	8
图 1-3	RowGenerator IP	9
图 1-4	WindowGenerator IP	9
图 1-5	MeanFilter IP	10
图 2-1	复制一个实验 1 副本.....	12
图 2-2	重命名实验目录.....	12
图 2-3	启动 Vivado.....	13
图 2-4	打开工程.....	13
图 2-5	设置：添加 IP 库.....	14
图 2-6	选择图像 IP 库.....	14
图 2-7	图像处理 IP 库中的 IP 列表显示	15
图 2-8	实验初始视图.....	16
图 2-9	添加灰度化 IP.....	16
图 2-10	断开 DVI2RGB IP 的 RGB 端口	17
图 2-11	断开端口后视图.....	17
图 2-12	添加 RowsGenerator IP	18
图 2-13	添加 WindowGenerator IP	18
图 2-14	添加 MeanFilter IP	19
图 2-15	添加 4 个 IP 后视图	19
图 2-16	Graying IP 配置	20
图 2-17	RowsGenerator IP 配置	20
图 2-18	WindowGenerator IP 配置	21
图 2-19	MeanFilter IP 配置.....	22
图 2-20	连接端口.....	22
图 2-21	连接 IP 后的视图	23
图 2-22	添加 Concat IP	24
图 2-23	添加 Constant IP 后的视图	24
图 2-24	Concat IP 设置	25
图 2-25	端口连接检查.....	25
图 2-26	保存设计.....	26




	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	3 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

图 2-27	创建实验顶层 Wrapper 文件	26
图 2-28	自动更新顶层文件.....	26
图 2-29	Generate Bitstream.....	27
图 2-30	点击 Yes 确认生成 bit 文件	27
图 2-31	打开 Hardware Manager	27
图 2-32	硬件连接对应位置.....	28
图 2-33	实际硬件连接.....	28
图 2-34	Open target	29
图 2-35	Program Device	29
图 2-36	烧写目标器件.....	30
图 2-37	编程进度条.....	30
图 3-1	均值滤波显示结果.....	31

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	4 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

表目录

表 1	RowGenerator IP 端口列表	9
表 2	WindowGenerator IP 端口列表.....	10
表 3	MeanFilter IP 端口列表	10

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	5 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

1. 实验简介

该实验通过设定一定尺寸的滑动窗口对图像的像素数据进行平均值计算，从而得到一个整体平滑的图像，滑动窗口最大为 15x15。

- **对于初学者，整个实验预计耗时 1 小时。**
- **对于进阶者，整个实验预计耗时 10 分钟。**

1.1 概述

均值滤波是典型的线性滤波算法，它是指在图像上对目标像素给一个模板，该模板包括了其周围的临近像素（以目标像素为中心的周围 8 个像素，构成一个滤波模板，即去掉目标像素本身），再用模板中的全体像素的平均值来代替原来像素值。

均值滤波器的概念非常的直观，使用滤波器窗口内的像素的平均灰度值代替图像中的像素值，这样的结果就是降低图像中的“尖锐”变化。这就造成，均值滤波器可以降低噪声的同时，也会模糊图像的边缘。均值滤波器的处理结果是过滤掉图像中的“不相关”细节，其中“不相关”细节指的是：与滤波器模板尺寸相比较小的像素区域。

根据均值计算方法的不同，均值滤波器有以下几种：

- 算术均值滤波器
- 几何均值滤波器
- 谐波均值滤波器
- 逆谐波均值滤波器


本实验采用的是算术均值滤波器，这是最简单的均值滤波器，可以去除均匀噪声和高斯噪声，但会对图像造成一定程度的模糊。

令 S_{xy} 表示中心点在 (x,y) 处，大小为 $m \times n$ 的滤波器窗口。算术均值滤波器就是简单的计算窗口区域的像素均值，然后将均值赋值给窗口中心点处的像素：

$$f(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

以 3×3 为例窗口模板如下所示：

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	6 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

1.2 实验目标

本实验的目标为 SWORD4.0 能够对 HDMI 输入的带有椒盐噪声的图像进行均值滤波后在显示器上输出无噪声的视频画面。


1.3 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式计算机 HDMI 输出/带 HDMI 输出的视频机顶盒
	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
软件	Vivado Design Suite	1	版本: 2014.4
	视频接口 IP 库	1	FPGA-Image-Library.zip*

*注: FPGA-Image-Library 为戴天宇开发的一个开源图像处理 IP 库, 该 IP 库遵循 LGPL, 详情请见: <http://fil.dtysky.moe>

1.4 实验原理

该实验的连接方式如下图所示:

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	7 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

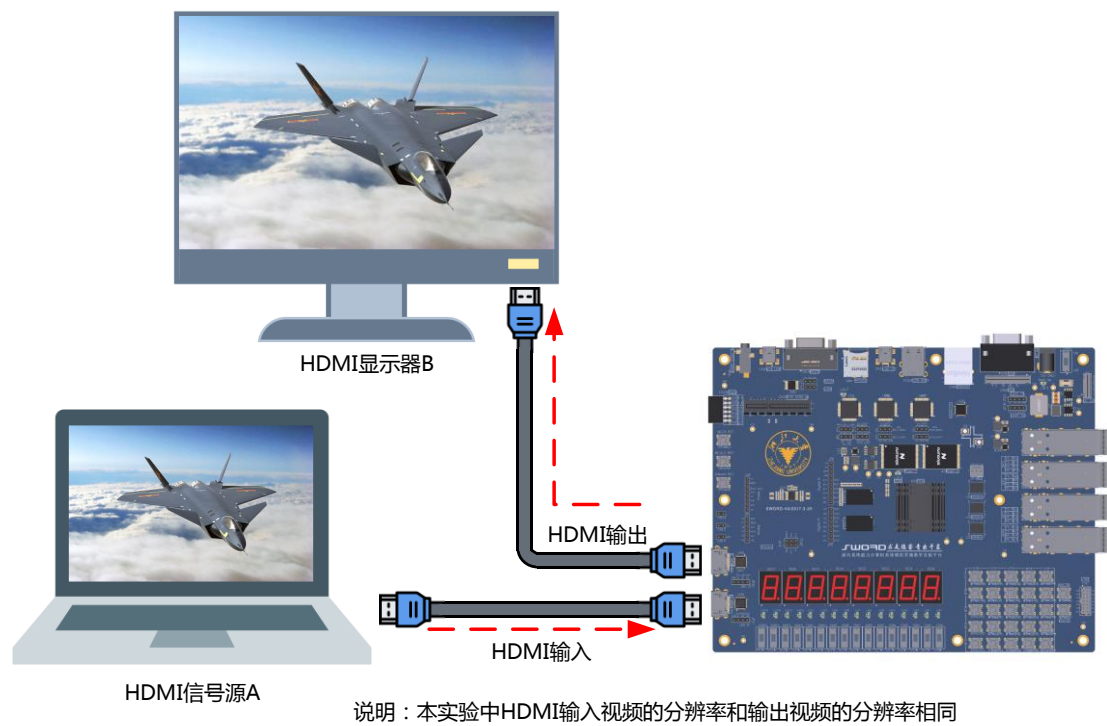


图 1-1 实验连接示意图

实验利用了 1 个 IP 来实现对比度变换：ContrastTransform。其中：对比度变换实验 IP 连接示意图如下图所示：

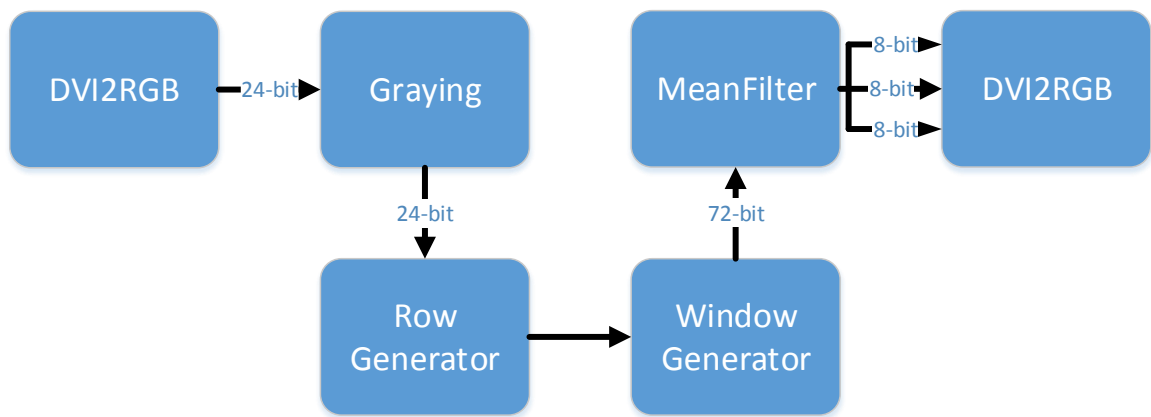


图 1-2 图像对比度调节连接示意图

RowGenerator 这个 IP 的作用是行缓存。

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	8 of 31
	作者	修改日期	公开	
Joseph Xu		2018/4/27		

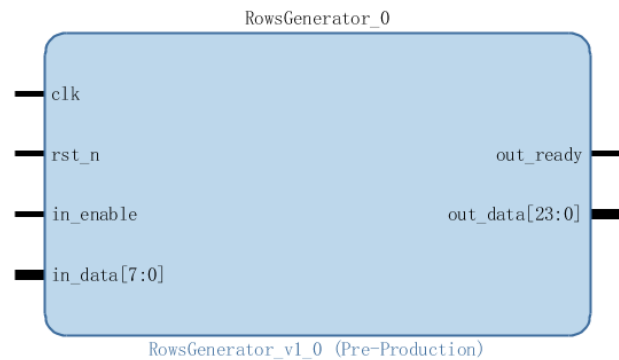


图 1-3 RowGenerator IP

该 IP 的端口信号定义如下表所示：

表 1 RowGenerator IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	I	1	Input data enable, it works as fifo0's wr_en.
in_data	I	Color_Width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	rows_width * color_width	Output data, it will be synchronous with out_ready.The lowest color_width-bits of this are the first row!

而 WindowGenerator 这个 IP 的作用是滑动窗口缓存。

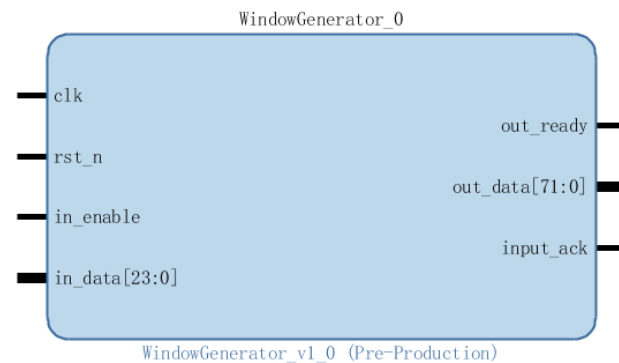


图 1-4 WindowGenerator IP

该 IP 的端口信号定义如下表所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	9 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

表 2 WindowGenerator IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	O	1	Input data enable, in pipeline mode, it works as another rst_n, in req-ack mode, only it is high will in_data can be really changes.
in_data	O	color_width * window_width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	color_width* window_width *window_width	Output data, it will be synchronous with out_ready.
input_ack	O	1	Input ack, only used for req-ack mode, this port will give a ack while the input_data received.

而 MeanFilter 这个 IP 的作用是（算术）均值滤波器。

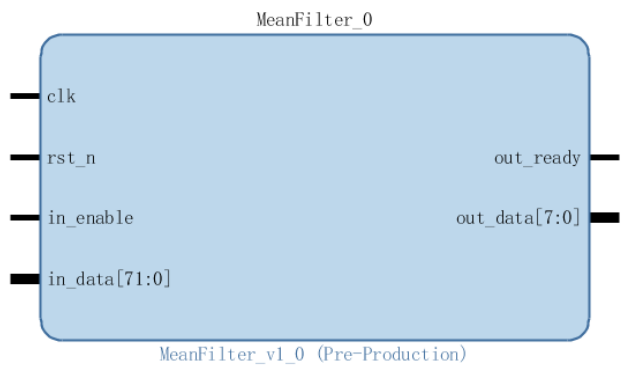



图 1-5 MeanFilter IP

该 IP 的端口信号定义如下表所示：

表 3 MeanFilter IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	I	1	Input data enable, in pipeline mode, it works as another rst_n, in req-ack

信号名	方向	宽度	含义
			mode, only it is high will in_data can be really changes.
in_data	I	color_width * window_width * window_width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	color_width	Output data, it will be synchronous with out_ready.

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	11 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

2. 均值滤波实验流程

本章将详细描述如何在 Vivado 2014.4 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。

2.1 操作步骤

- 1
- 由于本实验是在实验 1 的基础上进行扩展,所以我们将之前的实验部分复制 1 份,具体做法为在 D:\ImageLabs 文件夹下,将鼠标左键选中 lab1,然后按住 Ctrl 键不放,并拖拽到空白处,这样得到一个 lab1 的副本,如下图所示:

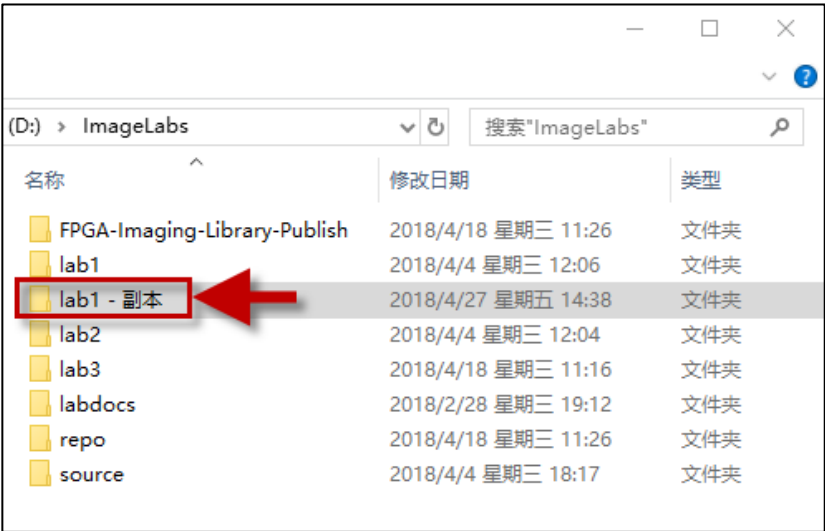


图 2-1 复制一个实验 1 副本

然后将 lab1 的副本重命名为 lab4,如下图所示,至此我们就可以在 lab4 文件夹里开始我们的实验内容:

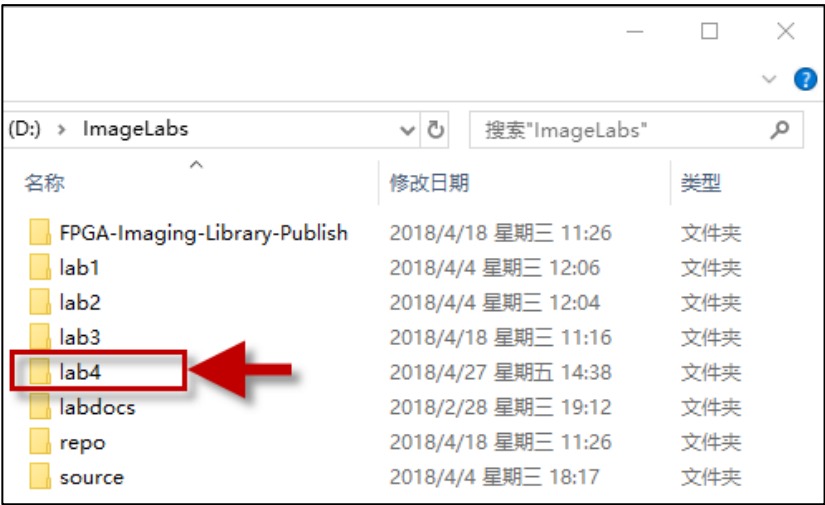



图 2-2 重命名实验目录

- 2
- 接着启动 Vivado 2014.4,在启动界面选择 Open Project,如下图所示:

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	12 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

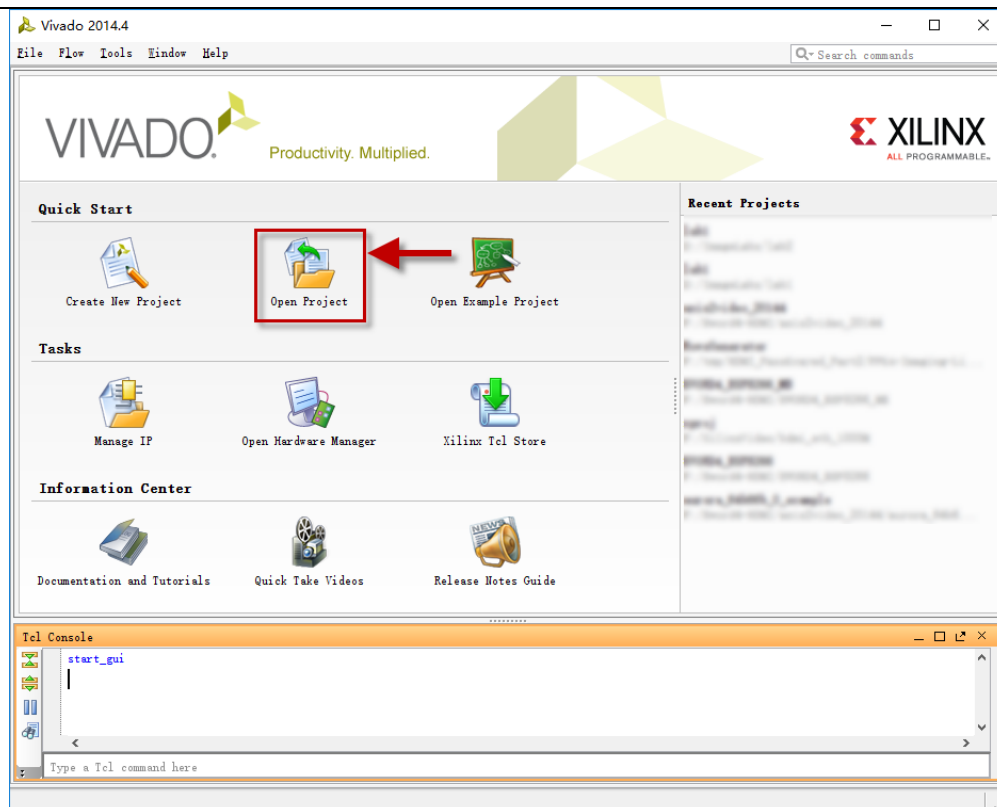


图 2-3 启动 Vivado

- 3 然后在选择对话框中 找到之前的 lab4(即 D:\ImageLabs\lab4) 然后选择 lab1.xpr 文件，点击 OK，打开工程，整个过程如下图所示：

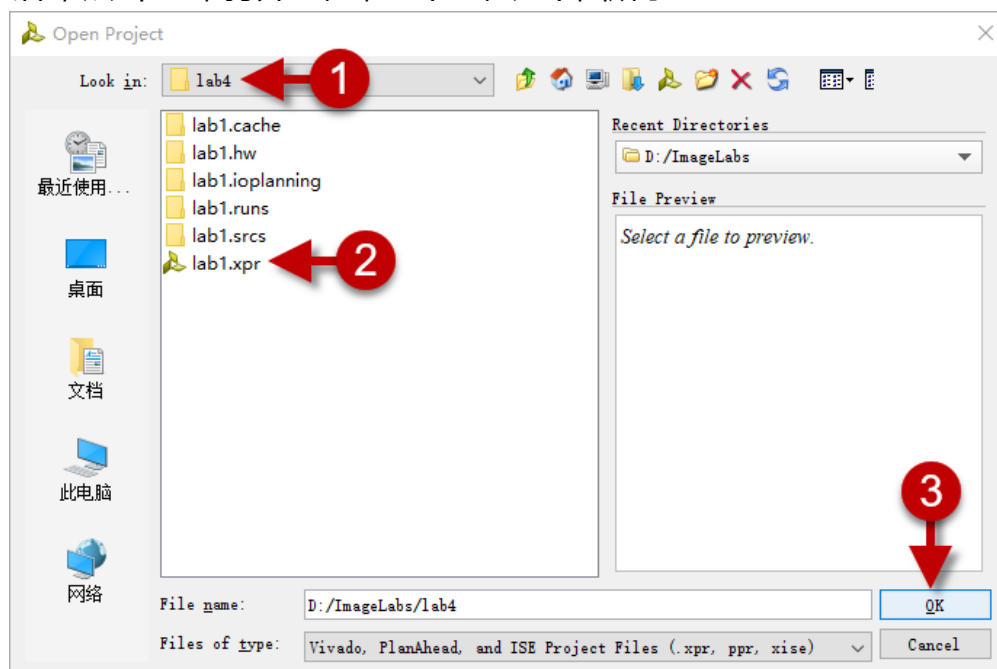



图 2-4 打开工程

- 4 接着我们将 FPGA-Image-Library IP 库添加进来，在 Vivado 的主界面点击 Project Settings，接着在新开对话框中点击 IP，点击 Add Repository，过程如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	13 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

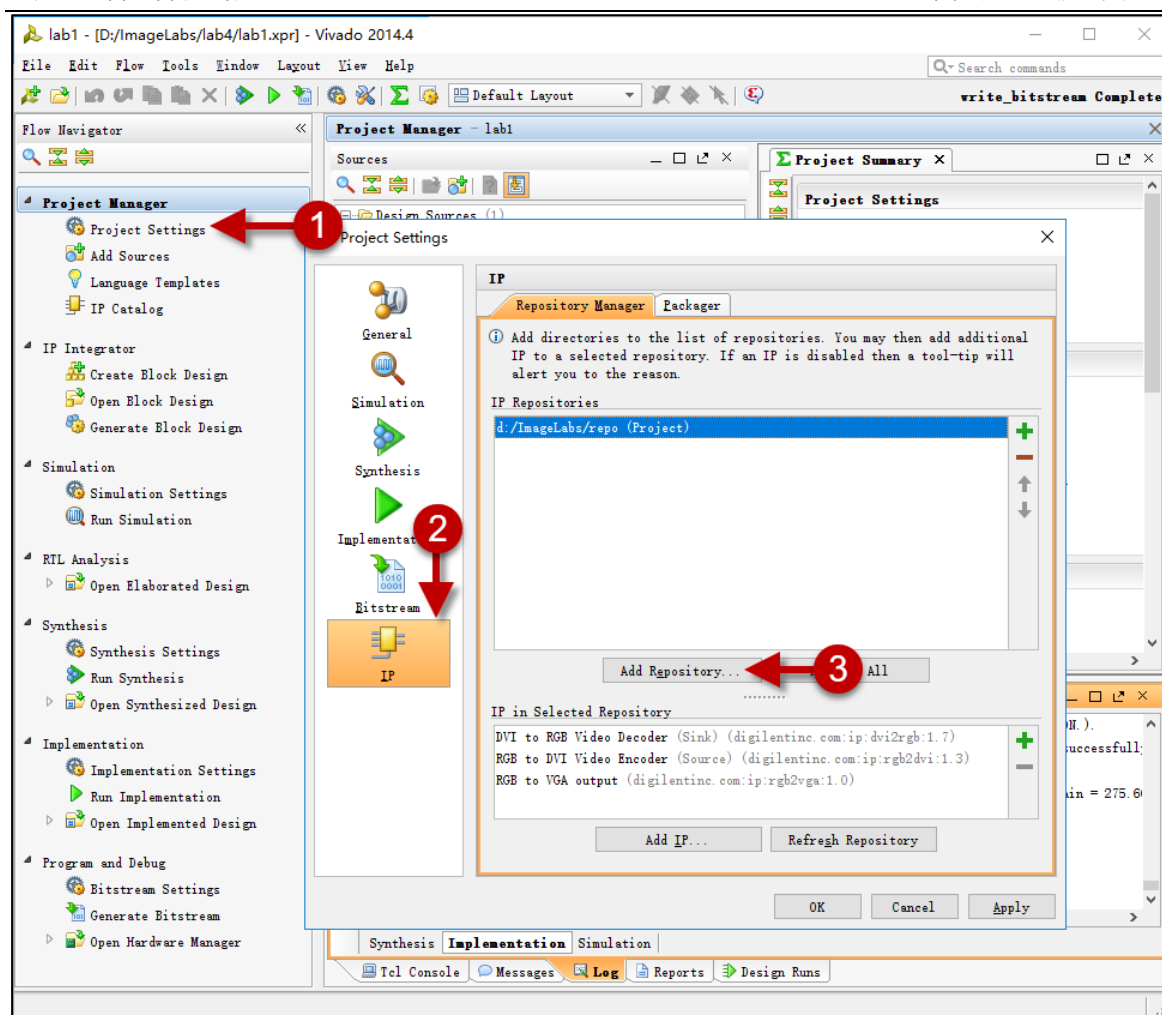


图 2-5 设置：添加 IP 库

- 5 在对话框中找到 D:\ImageLabs\FPGA-Image-Library-Publish , 然后点击 Select , 如下图所示：

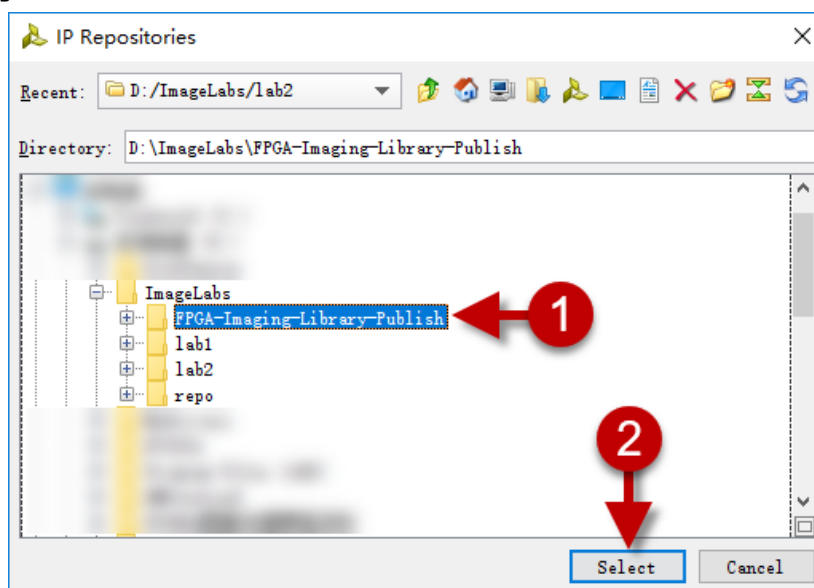



图 2-6 选择图像 IP 库

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	14 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

添加好 IP 库后，能看到 Vivado 会自动扫描库中的 IP，如果能看到如下图所示的一些 IP，则表示 IP 库添加成功，此时点击 OK 继续：

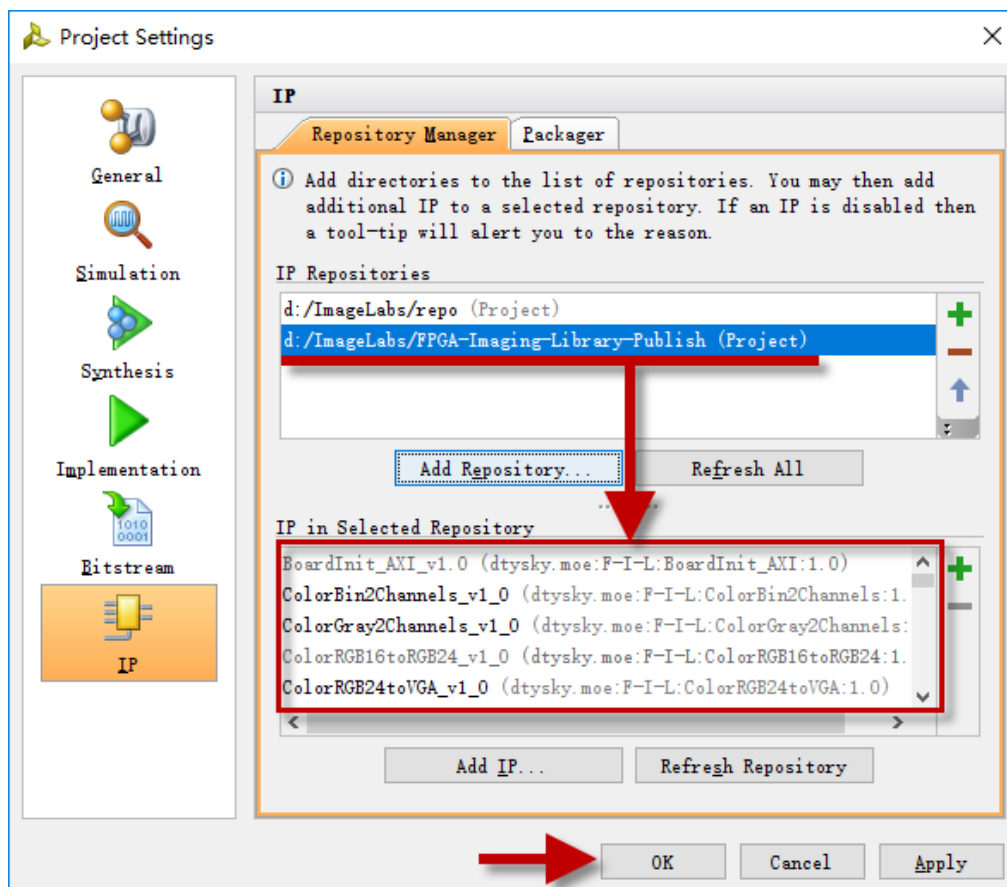



图 2-7 图像处理 IP 库中的 IP 列表显示

- 6 回到 Vivado 的主界面后，点击 Open Block Design，这时会在主界面右边区域看到之前实验 1 的 IP 结构，如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	15 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

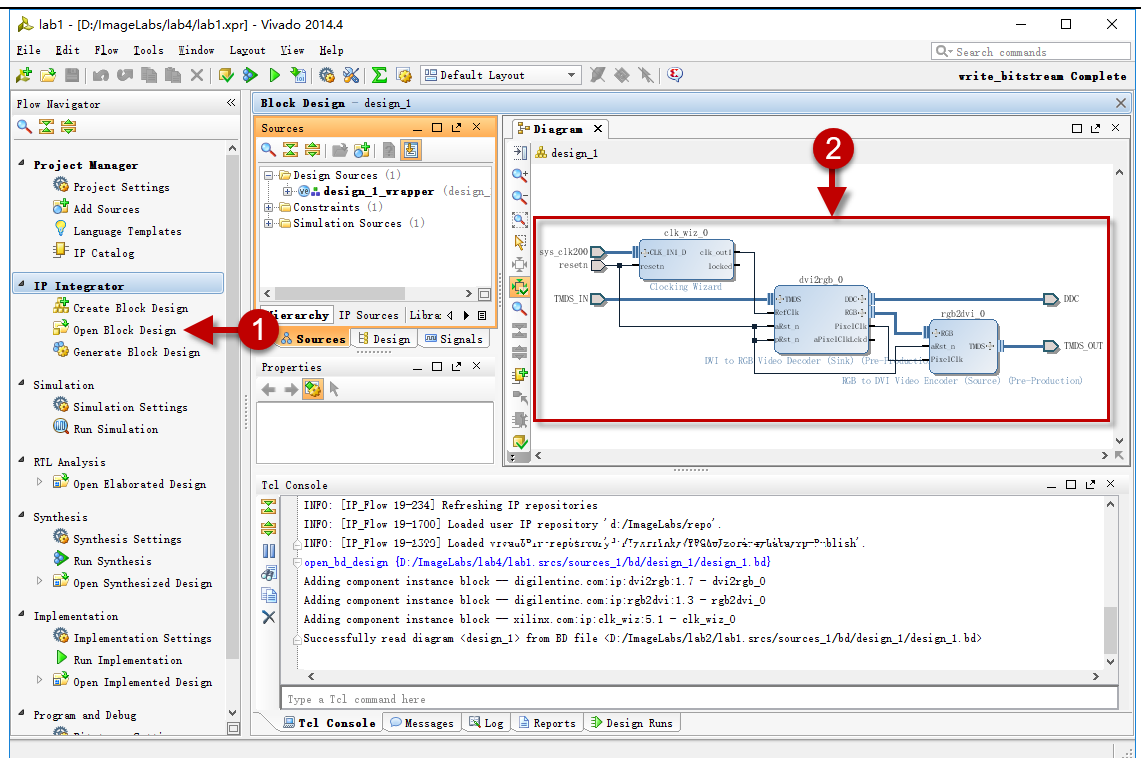


图 2-8 实验初始视图

- 7 在此基础上，我们开始添加 IP，点击左边栏的 Add IP 图标，然后在弹出的搜索框中，输入 gray，这时能看到搜索结果中有个 Graying 的 IP，双击它进行添加，整个过程如下图所示：

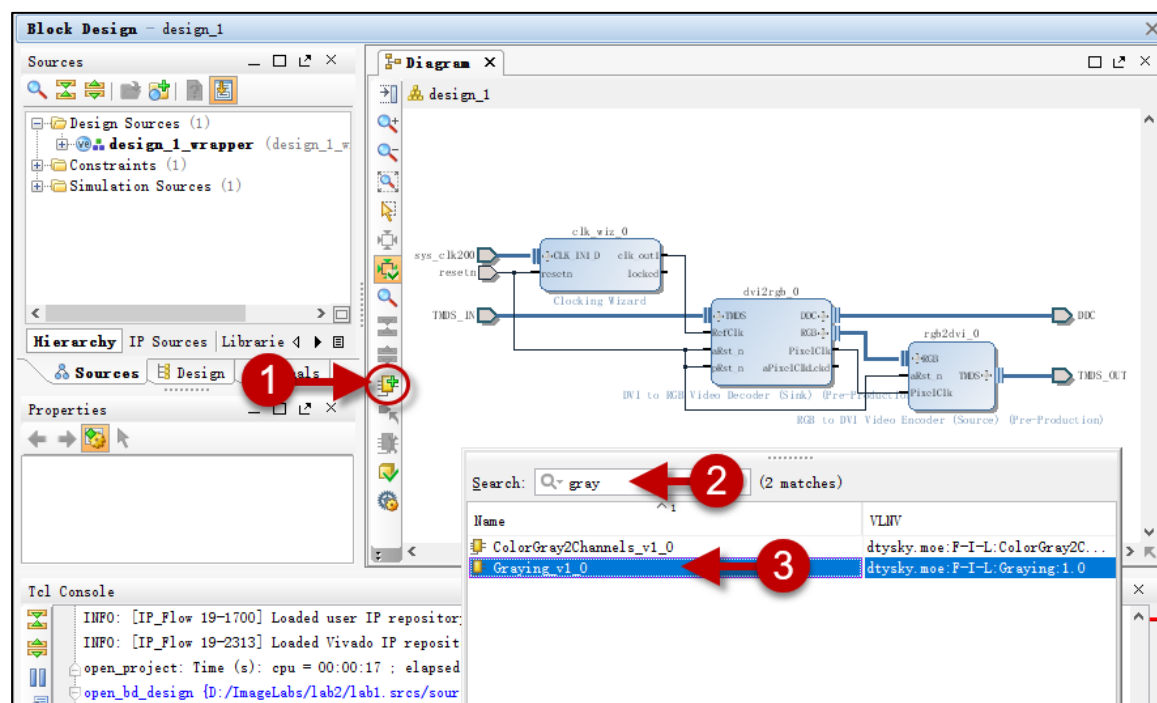



图 2-9 添加灰度化 IP

- 8 由于我们要将灰度化 IP 接入到图像数据流中，因此在模块化设计视图中，我们需要现将之前的数据流先断开，为此我们先用鼠标左键选中 DVI2RGB IP 的 RGB

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	16 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

端口,此时会看到 RGB 信号高亮为浅黄色(注意一定不要选中整个 IP),然后鼠标右键单击,在弹出菜单中选择 Disconnect Pin,整个过程如下图所示:

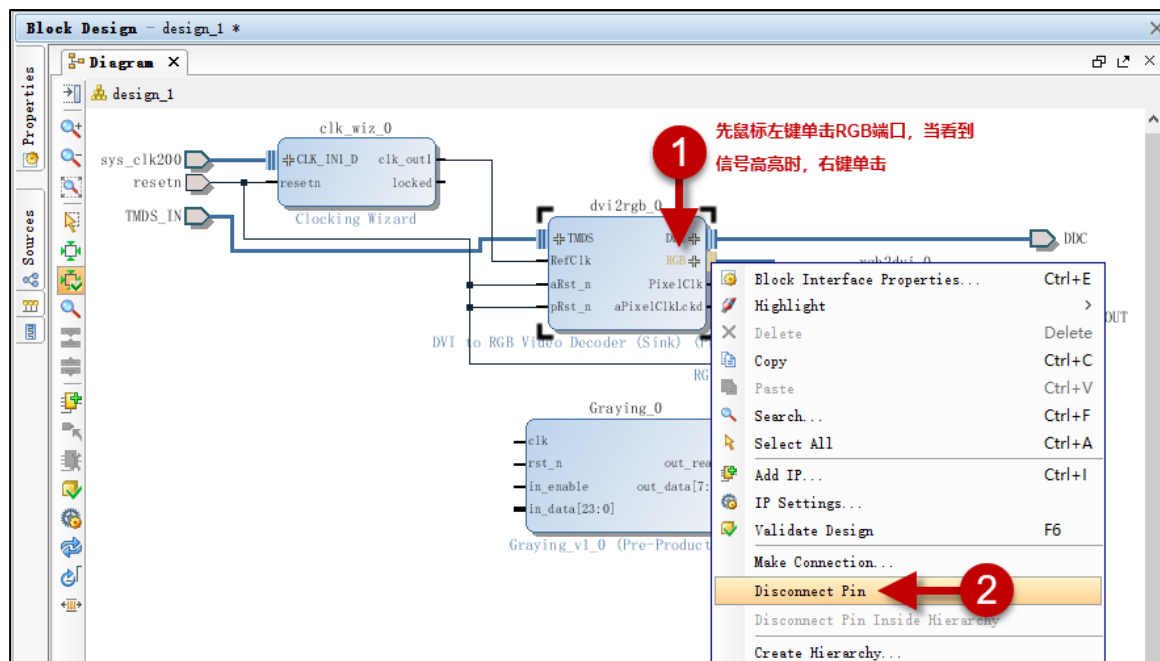


图 2-10 断开 DVI2RGB IP 的 RGB 端口

图像数据流断开后的 IP 视图如下图所示:

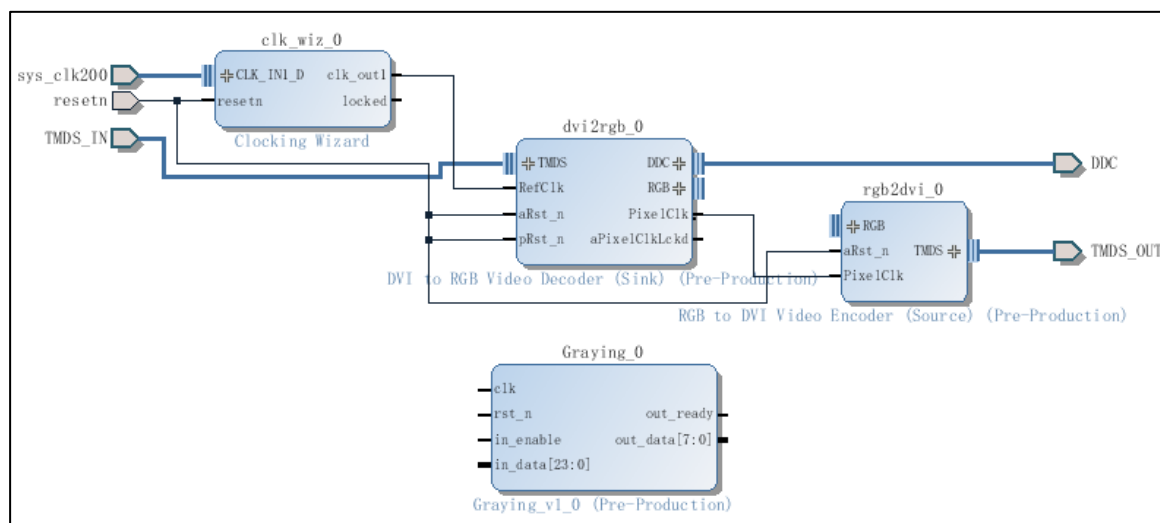



图 2-11 断开端口后视图

- 9 在此基础上,我们继续添加 IP,点击左边栏的 Add IP 图标,然后在弹出的搜索框中,输入 rows,这时能看到搜索结果中有个 RowsGenerator 的 IP,双击它进行添加,整个过程如下图所示:

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	17 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

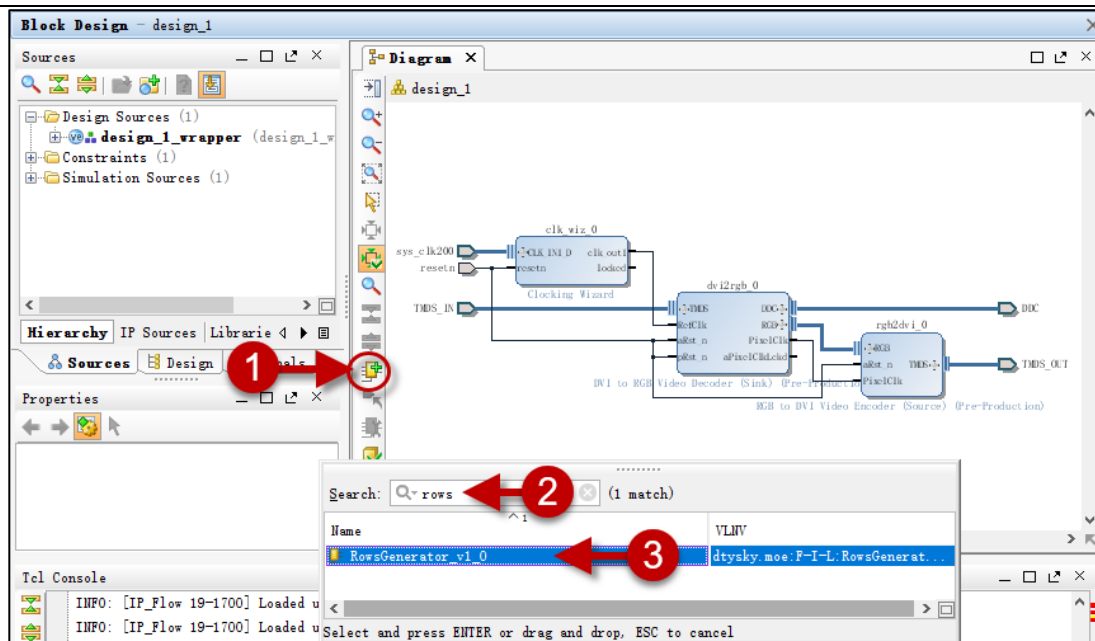


图 2-12 添加 RowsGenerator IP

接着继续添加 IP ,点击左边栏的 Add IP 图标 ,在弹出的搜索框中 ,输入 Window ,这时能看到搜索结果中有个 WindowGenerator 的 IP ,双击它进行添加 ,整个过程如下图所示 :

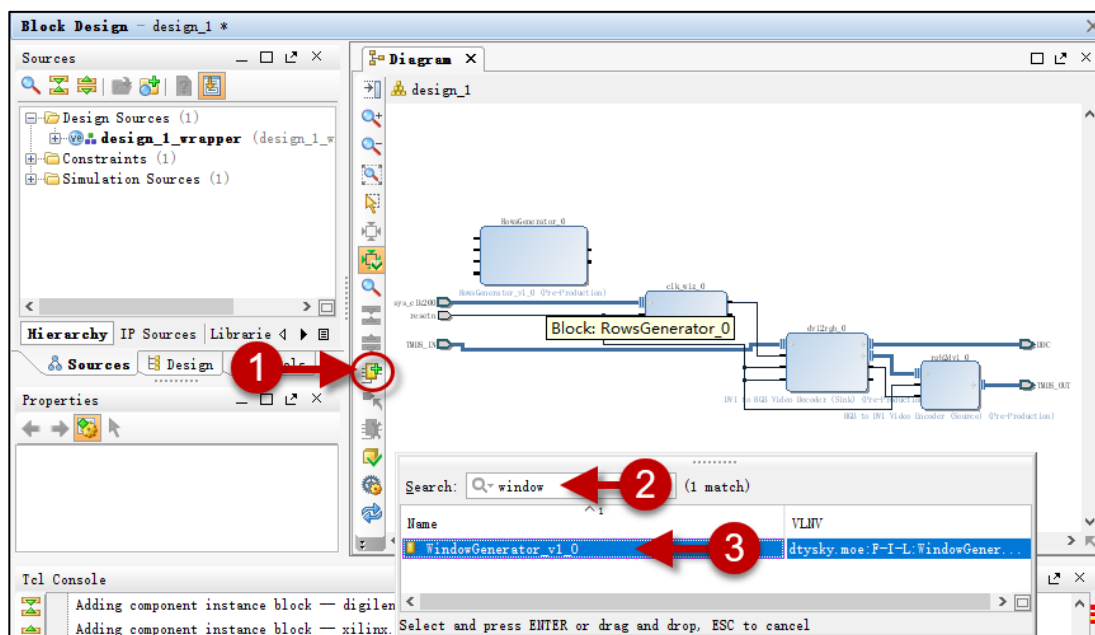



图 2-13 添加 WindowGenerator IP

接着继续添加 IP ,点击左边栏的 Add IP 图标 ,在弹出的搜索框中 ,输入 Mean ,这时能看到搜索结果中有个 MeanFilter 的 IP ,双击它进行添加 ,整个过程如下图所示 :

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	18 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

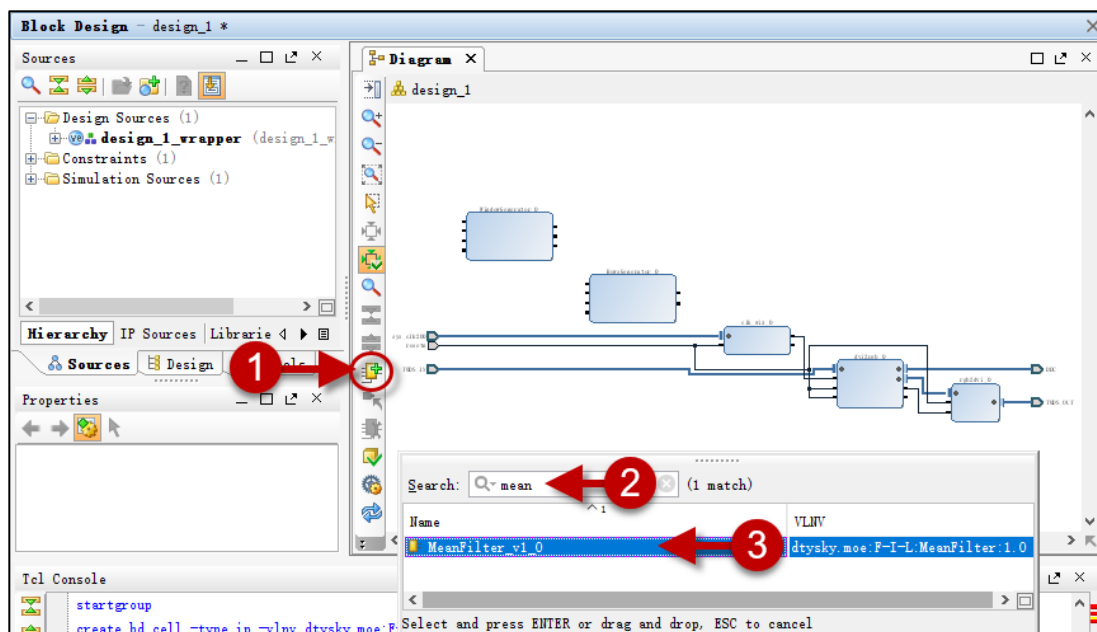


图 2-14 添加 MeanFilter IP

添加 4 个 IP 后的视图如下图所示：

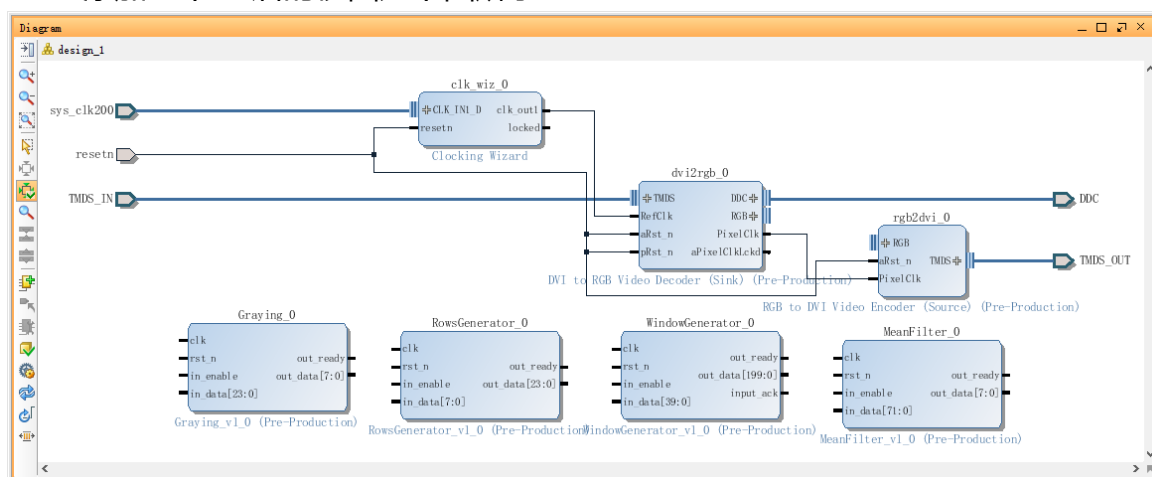



图 2-15 添加 4 个 IP 后视图

10 接着我们依次对这 4 个 IP 进行配置。

首先配置 Graying IP：双击 Graying_0 这个 IP，然后按照如下方式进行配置：

- Work Mode：Pipeline
- Color Width：8
- mul_delay：4

确认上述设置后，点击 OK 确定，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	19 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

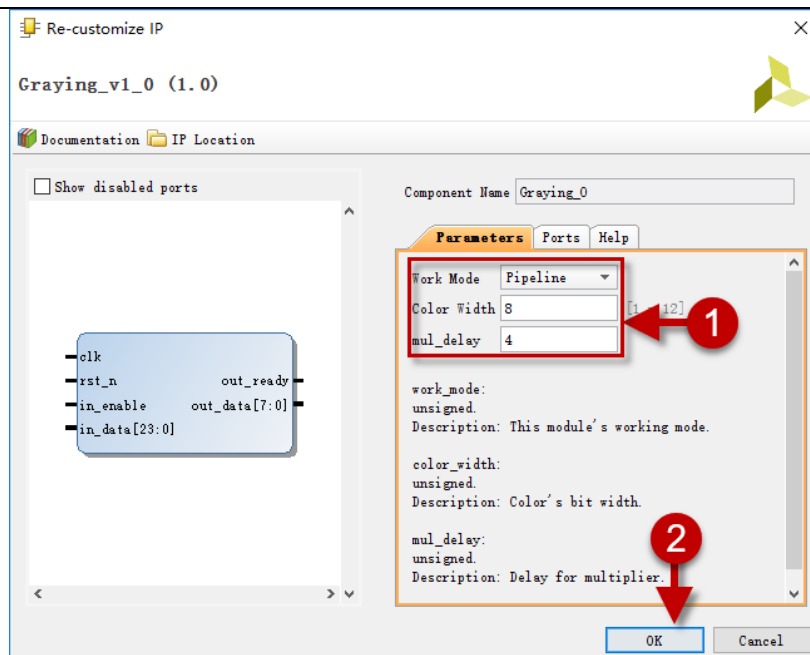


图 2-16 Graying IP 配置

接着配置 RowsGenerator IP，双击 RowsGenerator_0 这个 IP，然后按照如下方式进行配置：

- Rows Width：3
- Im Width：1274
- Color Width：8
- Im Width Bits：11

确认上述设置后，点击 OK 确定，整个过程如下图所示：

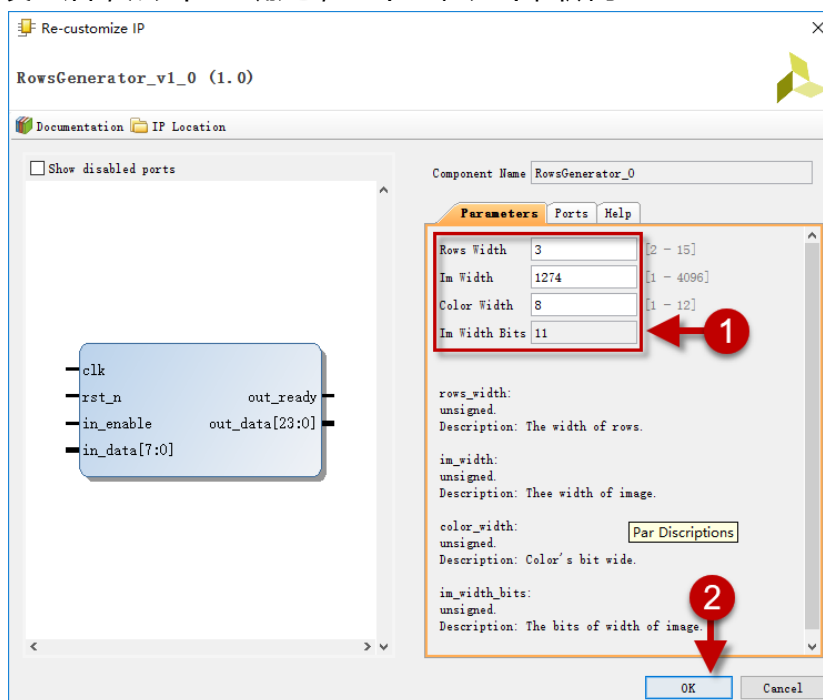



图 2-17 RowsGenerator IP 配置

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	20 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

接着配置 WindowGenerator IP , 双击 WindowGenerator_0 这个 IP , 然后按照如下方式进行配置 :

- Work Mode : Pipeline
- Window Width : 3
- Color Width : 8
- Window Width Half : 2

确认上述设置后 , 点击 OK 确定 , 整个过程如下图所示 :

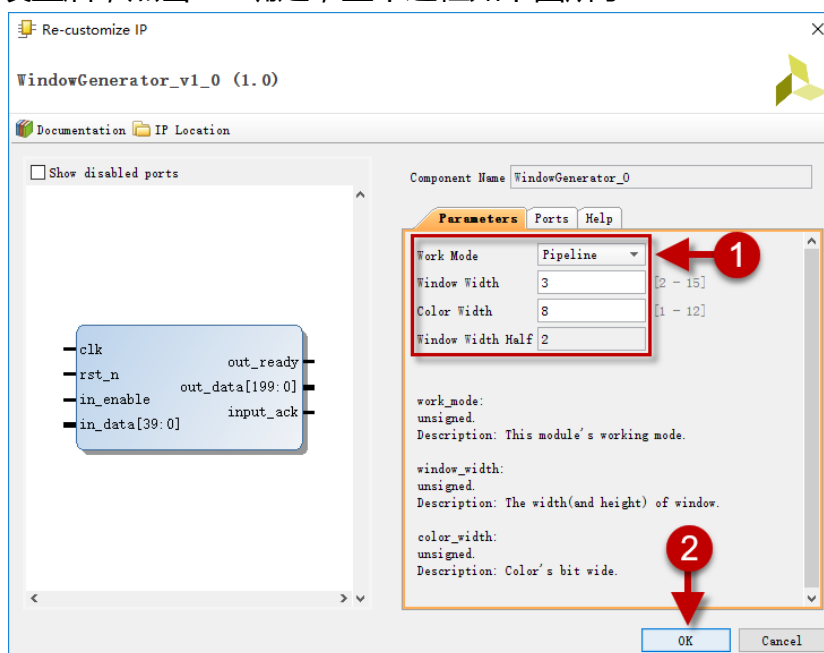



图 2-18 WindowGenerator IP 配置

最后配置 MeanFilter IP , 双击 MeanFilter_0 这个 IP , 然后按照如下方式进行配置 :

- Work Mode : Pipeline
- Window Width : 3
- Color Width : 8
- Sum Stage : 3

确认上述设置后 , 点击 OK 确定 , 整个过程如下图所示 :

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	21 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

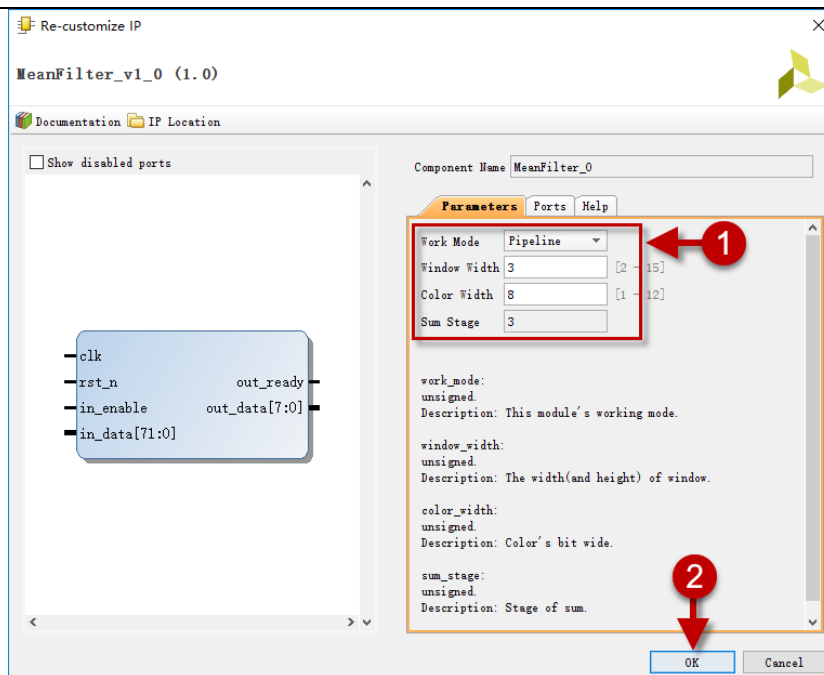


图 2-19 MeanFilter IP 配置

上述 IP 配置好后，我们将按照如下方式先对各个 IP 进行部分连接：

Graying_0:clk → dvi2rgb_0:PixelClk

Graying_0:rst_n → clk_wiz_0:resetn

Graying_0:in_enable → dvi2rgb_0:vid_pVDE

Graying_0:in_data[23:0] → dvi2rgb_0:vid_pData[23:0]

连接的操作如下图所示，鼠标光标移动到对应端口，然后点击不放并拖拽到与之对应的连接端口上，如果符合连接规则，端口会自动显示绿色的勾。

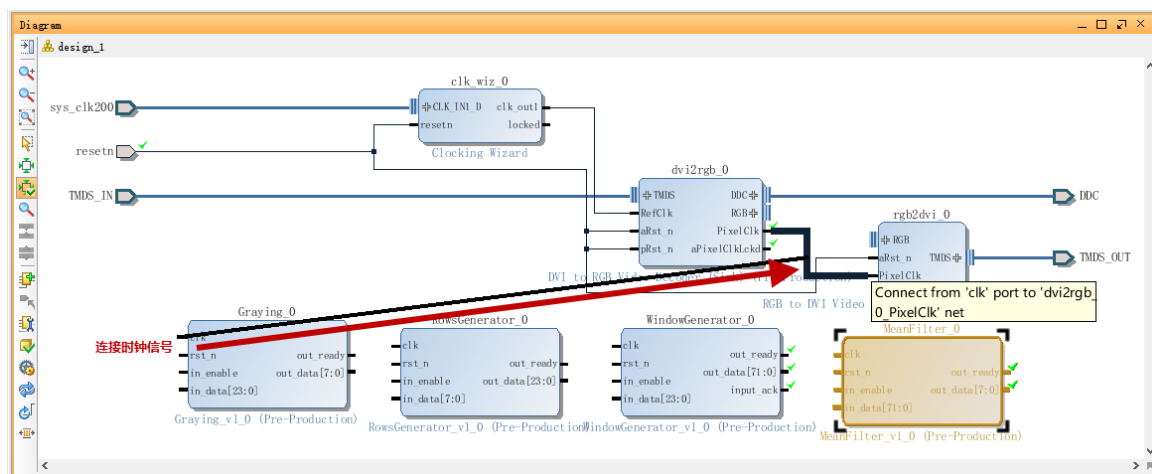


图 2-20 连接端口

- 11 之后将 Graying_0，RowsGenerator_0，WindowGenerator_0 和 MeanFilter_0，这 4 个 IP 进行连接，连接方式如下：

Graying_0:out_ready → RowsGenerator: in_enable

Graying_0:out_data[7:0] → RowsGenerator_0: in_data[7:0]

RowsGenerator: clk → dvi2rgb_0:PixelClk

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	22 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

RowsGenerator: rst_n → clk_wiz_0:resetn
 RowsGenerator: out_ready → WindowGenerator_0: in_enable
 RowsGenerator: out_data[23:0] → WindowGenerator_0: in_data[23:0]
 WindowGenerator_0: clk → dvi2rgb_0:PixelClk
 WindowGenerator_0: rst_n → clk_wiz_0:resetn
 WindowGenerator_0: out_ready → MeanFilter_0: in_enable
 WindowGenerator_0: out_data[71:0] → MeanFilter_0: in_data[71:0]
 MeanFilter_0: clk → dvi2rgb_0:PixelClk
 MeanFilter_0: rst_n → clk_wiz_0:resetn
 MeanFilter_0: out_ready → rgb2dvi_0:vid_pVDE
 dvi2rgb_0:vid_pHSync → rgb2dvi_0:vid_pHSync
 dvi2rgb_0:vid_pVSync → rgb2dvi_0:vid_pVSync

连接好后的 IP 视图如下图所示：

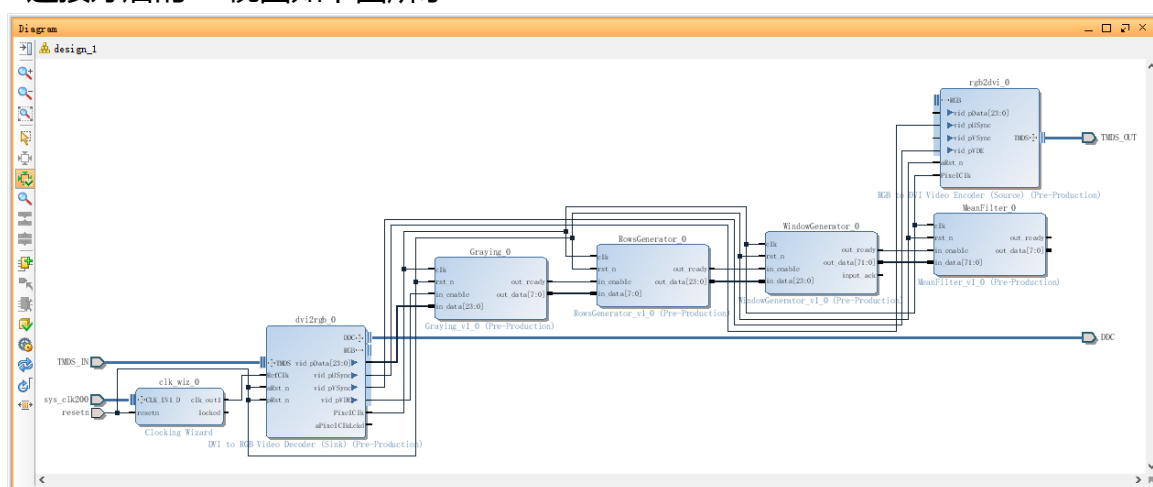



图 2-21 连接 IP 后的视图

- 12 由于均值滤波 IP 的输出是一个 8 位宽的像素数据，所以我们需要将其转换为一个 24 位宽像素数据。所以在本实验中，我们直接调用一个 concat 的 IP 来实现输入，点击 Add IP 图标，在弹出的搜索栏中输入 concat，然后在结果栏中双击 Concat 这个 IP 进行添加，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	23 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

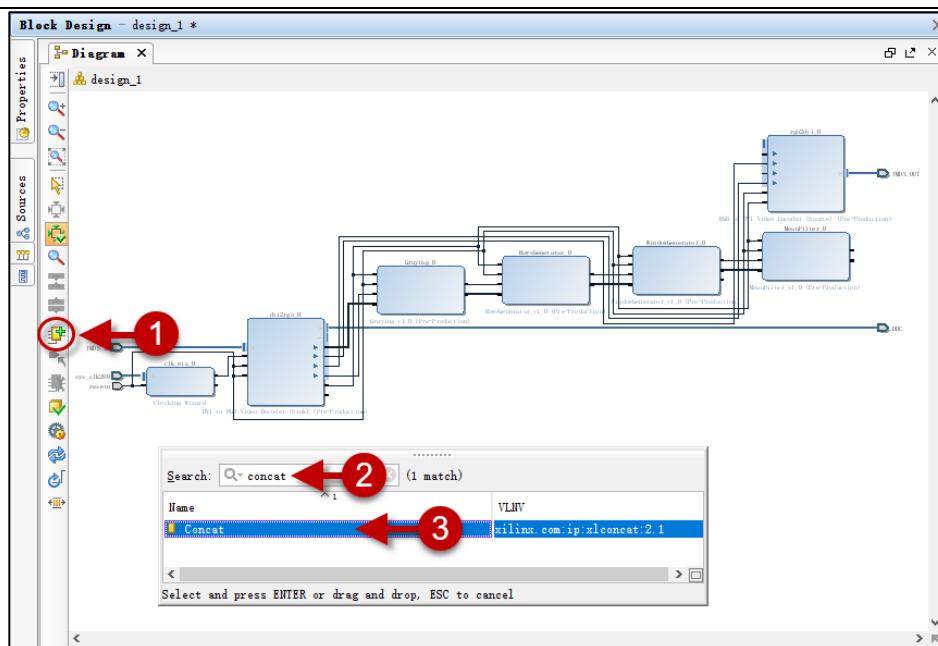


图 2-22 添加 Concat IP

添加后的局部 IP 视图如下图所示：

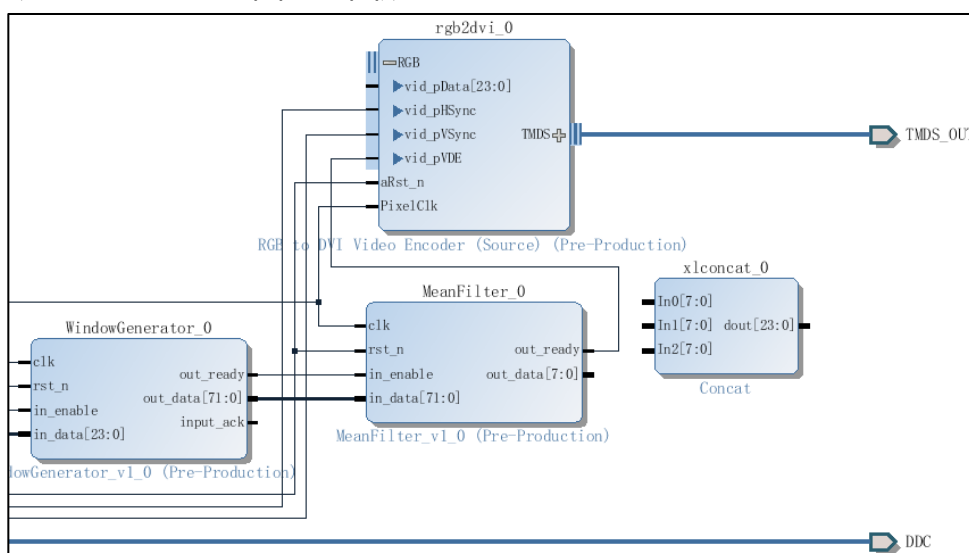


图 2-23 添加 Constant IP 后的视图

接着双击 xlconcat_0 这个 IP 进行配置，在配置对话框中，进行如下设置：

- Number of Ports : 3
- (Manual) In0 Width : 8
- (Manual) In1 Width : 8
- (Manual) In2 Width : 8

完成设置后，点击 OK 确认，整个拖成如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	24 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

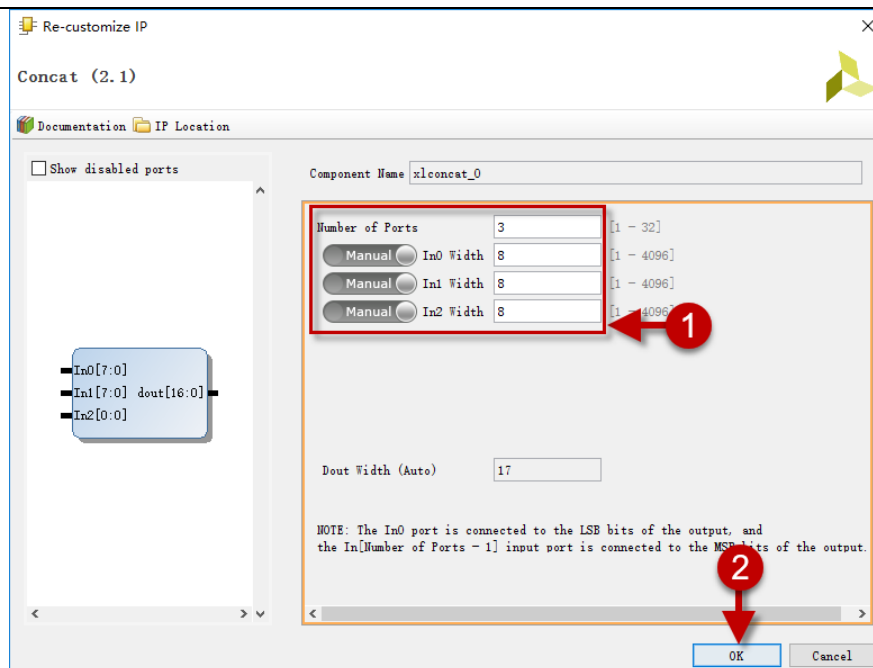


图 2-24 Concat IP 设置

- 13 请仔细检查各 IP 的端口连接是否正确,为了方便核对,本文特别准备了各种连接的高亮色图以示区别,如下图所示:

提示: 下图仅作为检查连接使用,读者完全不必也按照图示颜色进行标注 !!!

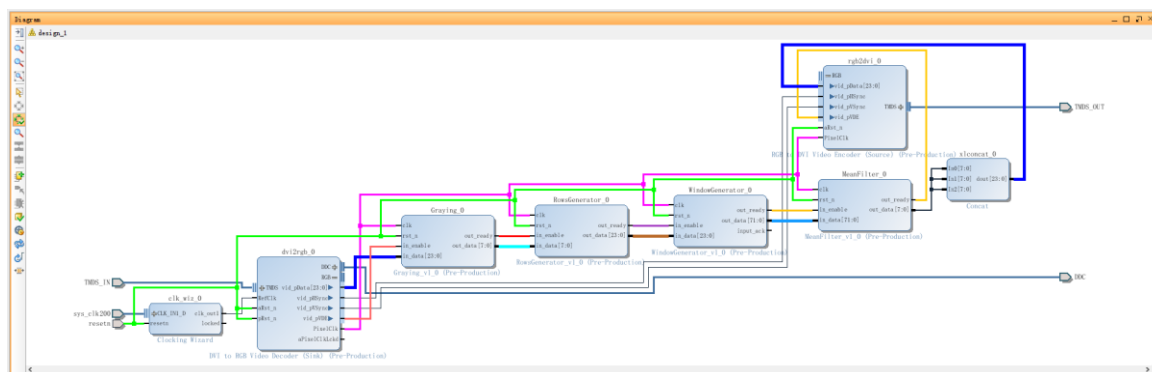


图 2-25 端口连接检查

- 14 连接检查无误后,即可保存 IP 模块化设计,在 Vivado 主界面点击保存图标,如下图所示:

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	25 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

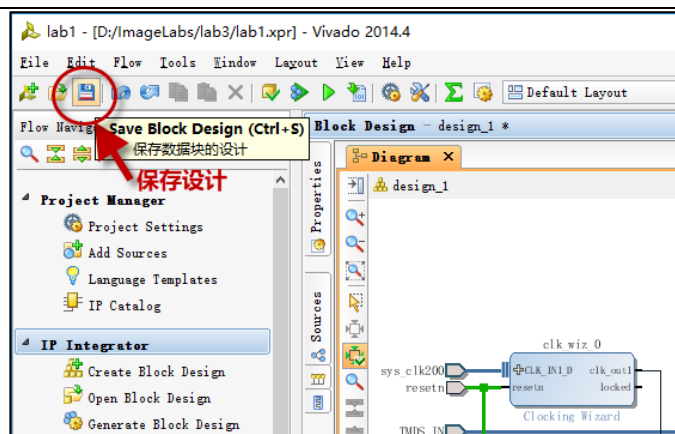


图 2-26 保存设计

接着在 Source 子窗口中展开 design_1_wrapper，选中 design_1.bd，鼠标右键单击，在弹出的菜单中选择 Create HDL Wrapper，整个过程如下图所示：

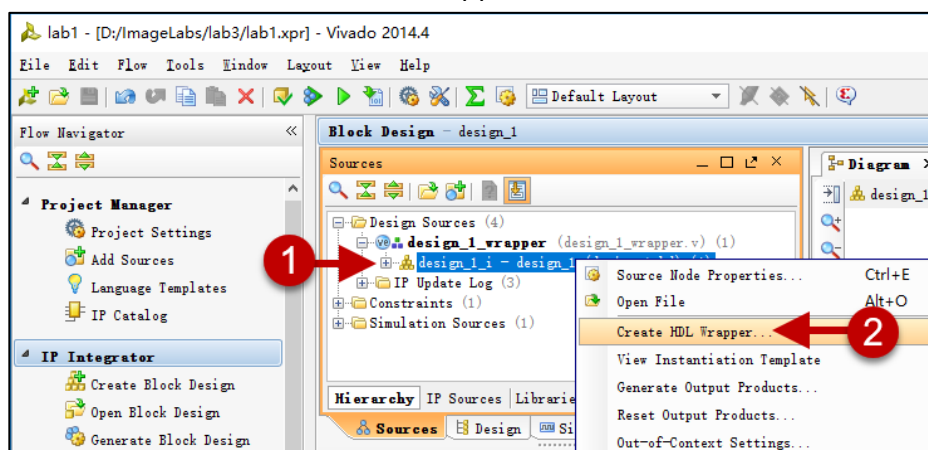


图 2-27 创建实验顶层 Wrapper 文件

接着在弹出的对话框中，保持默认的选项不变，即选择 Let Vivado manage wrapper and auto-update，然后点击 OK，如下图所示：

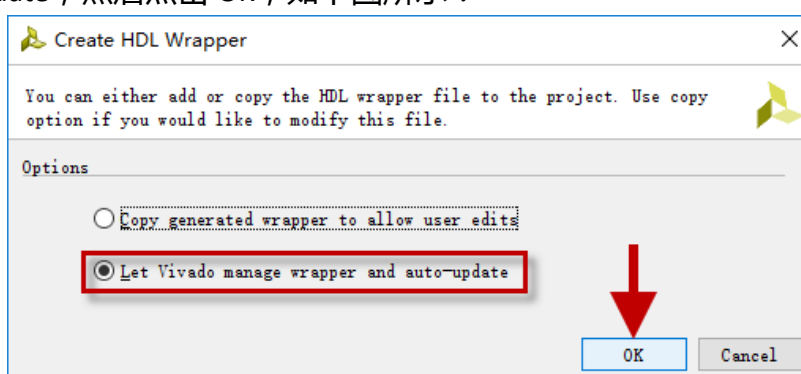


图 2-28 自动更新顶层文件

在 Vivado 主界面点击 Generate Bitstream，生成 bit 文件，如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	26 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

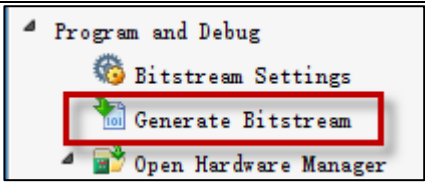


图 2-29 Generate Bitstream

在弹出的提示框中直接点 Yes 确认并继续，如下图所示：

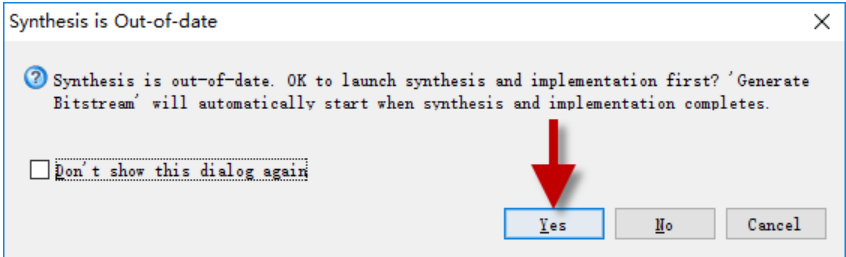


图 2-30 点击 Yes 确认生成 bit 文件

大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

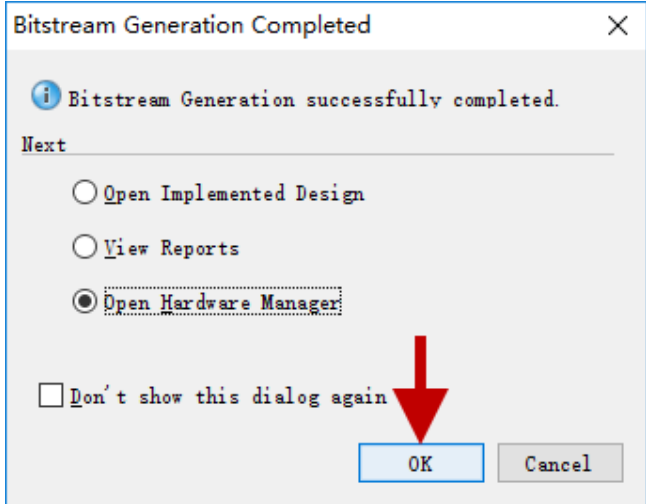


图 2-31 打开 Hardware Manager

接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	27 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

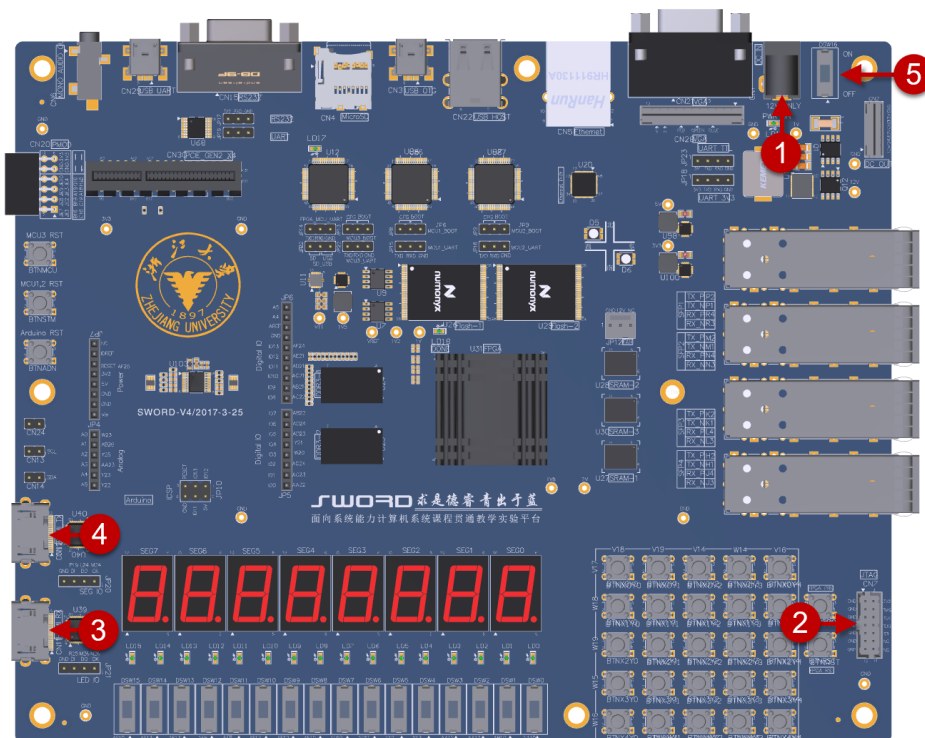



图 2-32 硬件连接对应位置

连接好后的效果如下图所示：



图 2-33 实际硬件连接

- 15 接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	28 of 31
	作者	修改日期	公开	
Joseph Xu		2018/4/27		

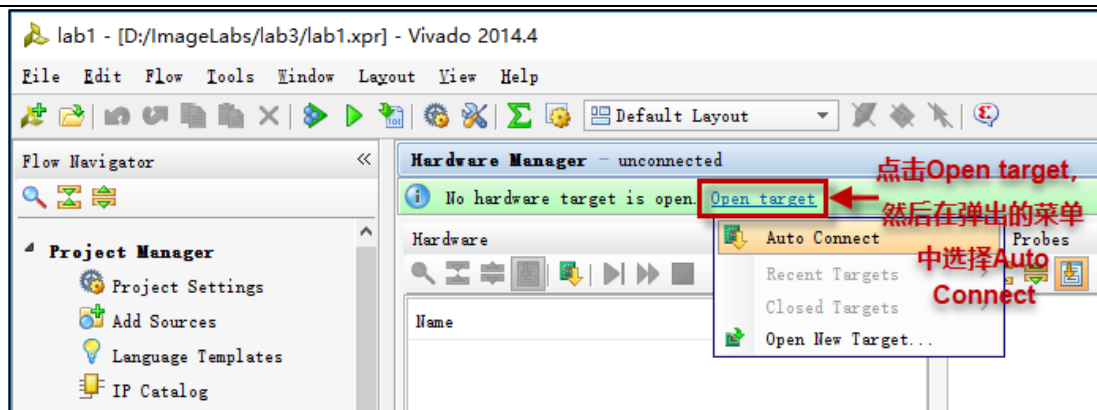


图 2-34 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

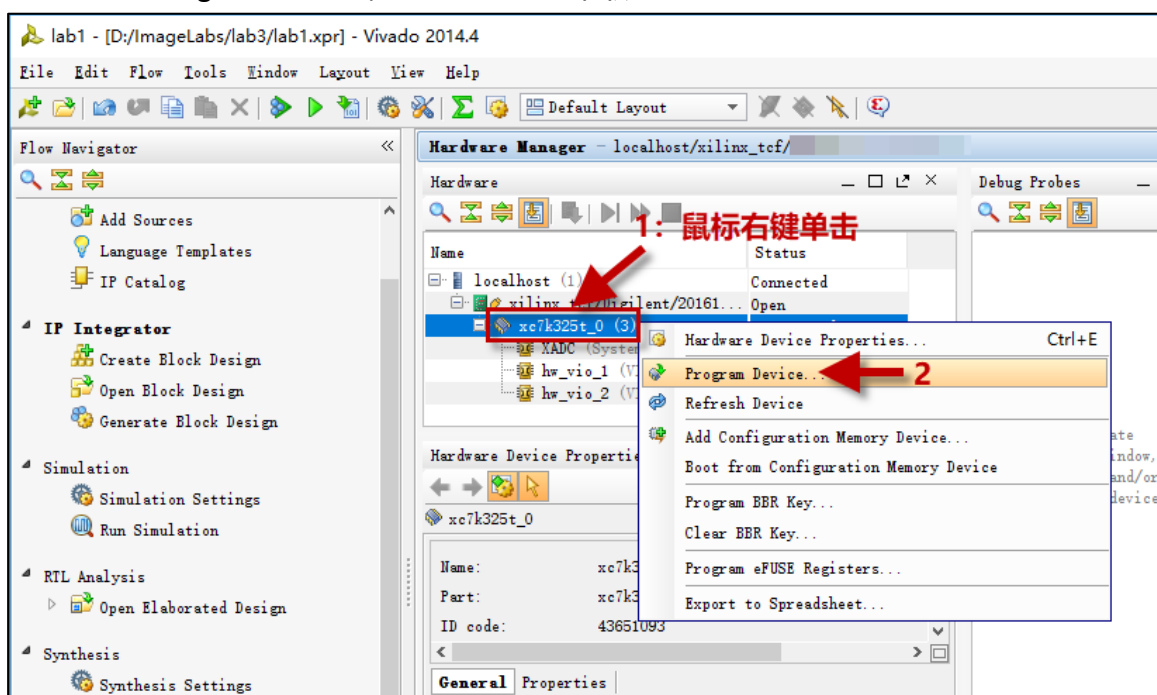


图 2-35 Program Device

在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	29 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

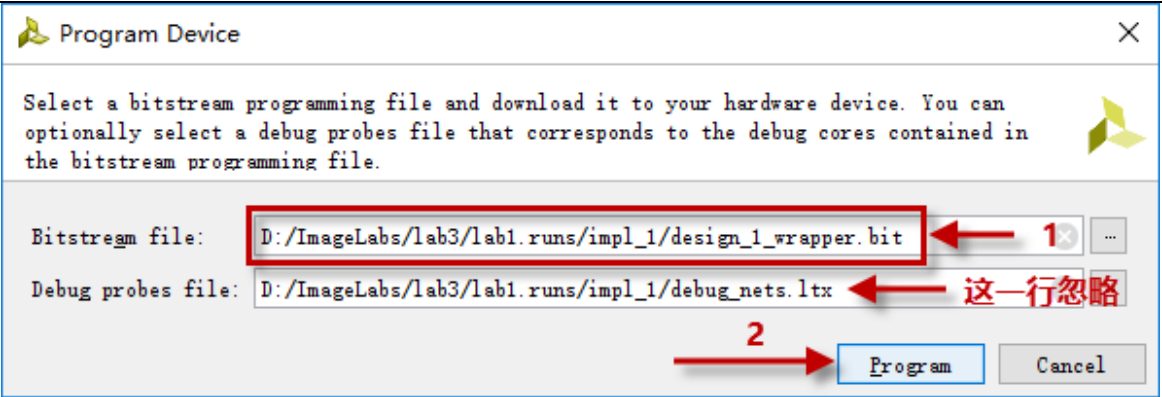


图 2-36 烧写目标器件

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

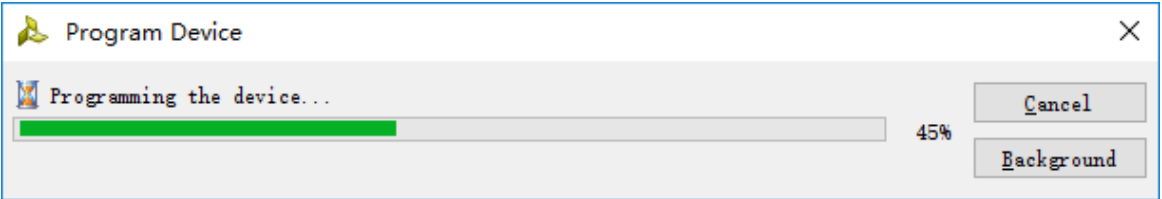


图 2-37 编程进度条

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	30 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		

3. 均值滤波实验结果

首先我们让 HDMI 信号源显示一幅带椒盐噪声的图片（位于同文件夹下的 J20-Salty.png），接着我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次，以便让信号源设备重新检测（Detect）一下接收设备，一切正常的话，我们即可在 HDMI 显示器上看到经过均值滤波后的显示画面。



图 3-1 均值滤波显示结果

	标题	文档编号	版本	页
	Lab4: 图像处理滤波器实验 1	XDI-SWORD-IMG-004	1.0	31 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/4/27		